Foresighted Robot Navigation in Human Environments

Maren Bennewitz
Why Foresighted Navigation?
Why Foresighted Navigation?

Oh, I’m stuck!
Problems to be Solved by Service Robots Acting in Human Environments

- Motion planning and navigation through cluttered regions
- Prediction of human motions for foresighted robot navigation
Navigation Through Cluttered Scenes

- **Reason about possible actions**

- **Complex actions** required
  - Whole-body motions
  - Object manipulation

- **Challenge:** Finding a detailed plan of all needed actions and whole-body motions is computationally expensive
Key Idea for Fast Planning

Exploit knowledge about **different obstacle classes** to choose appropriate robot actions
Our Approach - Overview

- **Segment objects** in the RGB image using a convolutional neural network (CNN)
- For each object class, **estimate the costs of possible actions** to overcome the obstacle based on execution time
- Encode the **action costs** corresponding to detected objects in a **2D cost grid** of the environment
- **Search for a 2D path** in this cost grid, which implicitly contains the necessary actions

[Regier et al. HUMANOIDS18, IJHR20]
Obtaining Class Information

Train a CNN for pixelwise object segmentation
### Actions for Obstacle Classes

<table>
<thead>
<tr>
<th>object class</th>
<th>action type</th>
</tr>
</thead>
<tbody>
<tr>
<td>balls</td>
<td>push, step over, pick up</td>
</tr>
<tr>
<td>cars, toy blocks</td>
<td>step over, pick up</td>
</tr>
<tr>
<td>stuffed toys, dolls</td>
<td>pick up</td>
</tr>
<tr>
<td>boxes, books</td>
<td>step onto, pick up</td>
</tr>
</tbody>
</table>
Actions for Obstacle Classes

- Use depth data to **exclude actions** depending on the **size of objects**
- Obstacles with **unknown class** or **no executable action** are considered as **static**
- Assign the “cheapest” possible action to each detected object
Action Costs = Time Needed

- Define the costs of actions according to the estimated completion time
- Learn the average execution time for each action from experimental runs
Cost Grid for Fast Path Planning

- **Project detected obstacles** with their action costs **onto a cost grid** of the environment
- Search for the **cheapest 2D path** to the goal location
- The obtained 2D path contains the **necessary actions**
Segmentation of the RGB Camera Image

robot camera view

doll

toy blocks

segmented RGB image
Projection on Cost Grid (Using Depth Data)
Path Planning in Cost Grid

- dolls
- toy blocks
- goal
- robot pose

- action cost
- darker color = higher cost

step over

robot pose

goal
Plan Execution

- Path segments **without objects**: walking control along the 2D path
- **Push/pickup**: walk to the last free grid cell on the 2D path and execute the corresponding action using the segmented point cloud
- **Step over/onto**: 3D footstep planning in the corresponding region on a height map computed from the point cloud
The Nao robot needs to reach the bottom part of the map.

For that it will need to navigate through the toy blocks or the stuffed animal.
Initially, our planner computes a path that suggests to perform the *step over* actions over the two blocks to be computed with a footstep planning algorithm.
Human Motion Prediction for Foresighted Robot Navigation
Human Motion Prediction

- To anticipate where service may be needed as well as to avoid interferences
- Observation: Humans typically move between objects and interact with them
Our Approach

- Learn transition probabilities for object interactions
- Use a map of the environment with the locations of relevant objects
- Integrate observations about the user and detected object interactions to predict their next navigation goal
- Realize foresighted robot navigation by choosing poses close to the user’s goal while avoiding interferences

[Bruckschen et al. RAS20, IROS20, RO-MAN20, ICSR19, ECMR19]
Learning Transitions of Object Interactions

- Extract general user-object interactions from RGB-D videos
- R-CNN trained on Open Images dataset for object detection
- OpenPose for the detection of users and estimation of their body poses

[Huang et al., CVPR17, Krasin et al., 17]  [Cao et al., CVPR17]
Detection of Interactions

- Based on objects the user is oriented to and that are touched
- Estimate the **user’s orientation** and the **hand positions** from the body pose
- Overlap between the **object bounding box** and the **hand position**, and check whether close in depth
Human-object interaction

Detected person, his pose, and objects
Prediction of the User’s Navigation Goal

- From the extracted human-object interactions, derive transition probabilities between objects.
- Use knowledge about object locations in the environment to infer likely navigation goals and predict human motion.

![Robot's view diagram with darker green indicating higher likelihood.](image)
Prediction of the User’s Navigation Goal

- Objects of a certain class can appear several times
- Integrate also observations about the user’s movements and their orientation to infer their navigation goal

![Diagram showing user and robot positions with darker green indicating higher likelihood]

darker green = higher likelihood
Prediction of the User’s Navigation Goal

- Objects of a certain class can appear several times
- Integrate also observations about the user’s movements and their orientation
- Update the likelihood of possible navigations goals based on
  - Inverse distance of the user to the objects and
  - Inverse orientation difference of the human towards the objects
Belief About the User’s Navigation Goal

\[ p(o_i | S) \]

object class + position

observed user position and pose
Belief About the User’s Navigation Goal

\[ p(o_i | S) = \eta \cdot p(S | o_i) p(o_i) \]
Belief About the User’s Navigation Goal

\[ p(o_i | S) \]

\[ = \eta \cdot p(S | o_i) p(o_i) \]

\[ = \eta \cdot p(S | o_i) I(o_i | \tau) \]

- Object class + position
- Observed user position and pose
- Normalizer
- Learned interaction transitions
- Observed human-object interaction
Observation Likelihood

\[ p(S|o_i) = \frac{1}{\text{dist}(x_h, x_i)} \cdot \frac{1}{\Delta(\theta_h, \theta_{opt})} \]
Belief about the User’s Navigation Goal

\[ p(o_i | S) \]
\[ = \eta \cdot p(S | o_i)p(o_i) \]
\[ = \eta \cdot p(S | o_i)I(o_i | \tau) \]
Evaluation Data Sets (1)

- 25 simulated office/home environments with sizes between 100-150m²
- Recording of trajectories by users who generated typical movements between objects
- For training the interaction model: dataset with 161 sequences containing several subsequent object transitions
For the quantitative evaluation of the navigation goal prediction, separate dataset of 64 trajectories between two objects

Generate an observation once every second as long as the human is moving

Evaluation with and without knowledge about the last object interaction
Evaluation of the Prediction

- After each observation, check whether the predicted most likely goal corresponds to the user’s true navigation goal.
- If so, count the prediction as correct.
- **Prediction accuracy**: Number of correct predictions / total number of observations.
- **Average position** on the user’s trajectory from which our system constantly returns the correct navigation goal.
Results: Navigation Goal Prediction

Dataset of 64 human trajectories collected in 25 simulated office/home environments

<table>
<thead>
<tr>
<th>Last interaction observed</th>
<th>Avg. Prediction Accuracy</th>
<th>Avg. Trajectory Length Until Correct Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last interaction not observed</td>
<td>0.48</td>
<td>52%</td>
</tr>
</tbody>
</table>
Foresighted Robot Positioning

- A robot needs to place itself so as to be close to the user in case service is needed
- Evaluate positions for the robot using a cost function
- Take into account the distance to the user’s possible navigation goals weighted by their probabilities
Computing the Best Robot Position

- Weigh robot positions
- Lower cost if close to likely navigation goals

\[ C_{\text{dist}}(X) = \sum_{o_j \in O} P(o_j|S) \cdot L(P_{X \rightarrow x_{o_j}}) \]

- Costs of 2D map cells
- Length of A* path from the cell to possible navigation goal
- Probability of possible next navigation goal
- Darker color = lower cost
- Best position
Path Planning Considering Human Comfort Constraints

- Robot should not enter:
  - Social zone (SZ)
  - Information process space (IPS)
- Increased costs: Area behind the human (B)

[Hall et al. 1968; Kitazawa et al. 2010]
Time-Dependent Path Planning

- Determine the **best robot position** using the belief of the user’s navigation goal
- **Predict user path** to their most likely navigation goal
- Apply **time-dependent path planning** to the best robot position considering **human comfort constraints**
Based on observations and prior knowledge about object transitions the robot predicts the movement of the user. Objects are colored in green if they are likely navigation goals. The darker the green the higher the likelihood.
Recent Work

- Predict a **sequence** of navigation goals of the user and infer where assistance is needed
- Integrate **context information** (e.g., room, activity) to choose best robot path
- Include general user-desired factors such as **unobtrusiveness** and **predictability** when calculating the robot’s path

[Bruckschen, Bungert et al. IROS20, RO-MAN20, RO-MAN21]
Summary (1)

- Foresighted robot navigation in human environments
- Exploit **knowledge about obstacle classes** during path planning through cluttered regions
- CNN to distinguish different obstacle classes
- Construct a **2D grid** that encodes the associated **action costs** derived from execution time
- Compute the robot’s path, which **implicitly contains all necessary actions** to handle the objects
Summary (2)

- Predict human motions and anticipate where service may be needed
- Learn sequences of human-object interactions from RGB-D data
- Utilize the learned object transitions to infer the user’s navigation goal and predict the corresponding path
- Realize foresighted robot navigation and placement based on that prediction
Literature

- Classifying Obstacles and Exploiting Class Information for Humanoid Navigation through Cluttered Environments
  P. Regier, A. Milioto, C. Stachniss, and M. Bennewitz
  International Journal of Humanoid Robotics (IJHR), 2020
Detection of Generic Human-Object Interactions in Video Streams
L. Bruckschen, S. Amft, J. Tanke, J. Gall, M. Bennewitz
Int. Conference on Social Robotics (ICSR), 2019

Predicting Human Navigation Goals based on Bayesian Inference and Activity Regions
L. Bruckschen, K. Bungert, N. Dengler, and M. Bennewitz
Robotics and Autonomous Systems (RAS), 2020

Human-Aware Robot Navigation by Long-Term Movement Prediction
L. Bruckschen, K. Bungert, N. Dengler, and M. Bennewitz
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020
Where Can I Help? Human-Aware Placement of Service Robots
IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2020

Human-Aware Robot Navigation Based on Learned Cost Values from User Studies
K. Bungert, L. Bruckschen, S. Krumpen, W. Rau, M. Weinmann, M. Bennewitz
IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2021